

Heat Map

April 30, 2018

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import folium
from folium.plugins import HeatMap
import seaborn as sns
from scipy import stats
```

```
In [2]: df = pd.read_csv("Crimes_-_2001_to_present.csv")
```

```
In [3]: df.head(10)
```

```
Out[3]:
```

	ID	Case Number	Date	Block	IUCR	\
0	10000092	HY189866	03/18/2015 07:44:00 PM	047XX W OHIO ST	041A	
1	10000094	HY190059	03/18/2015 11:00:00 PM	066XX S MARSHFIELD AVE	4625	
2	10000095	HY190052	03/18/2015 10:45:00 PM	044XX S LAKE PARK AVE	0486	
3	10000096	HY190054	03/18/2015 10:30:00 PM	051XX S MICHIGAN AVE	0460	
4	10000097	HY189976	03/18/2015 09:00:00 PM	047XX W ADAMS ST	031A	
5	10000098	HY190032	03/18/2015 10:00:00 PM	049XX S DREXEL BLVD	0460	
6	10000099	HY190047	03/18/2015 11:00:00 PM	070XX S MORGAN ST	0486	
7	10000100	HY189988	03/18/2015 09:35:00 PM	042XX S PRAIRIE AVE	0486	
8	10000101	HY190020	03/18/2015 10:09:00 PM	036XX S WOLCOTT AVE	1811	
9	10000104	HY189964	03/18/2015 09:25:00 PM	097XX S PRAIRIE AVE	0460	

	Primary Type	Description	Location Description	\
0	BATTERY	AGGRAVATED: HANDGUN	STREET	
1	OTHER OFFENSE	PAROLE VIOLATION	STREET	
2	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	
3	BATTERY	SIMPLE	APARTMENT	
4	ROBBERY	ARMED: HANDGUN	SIDEWALK	
5	BATTERY	SIMPLE	APARTMENT	
6	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	
7	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	
8	NARCOTICS	POSS: CANNABIS 30GMS OR LESS	STREET	
9	BATTERY	SIMPLE	RESIDENCE PORCH/HALLWAY	

	Arrest	Domestic	...	Ward	Community Area	\
0	False	False	...	28.0	25.0	
1	True	False	...	15.0	67.0	

2	False	True	...	4.0	39.0
3	False	False	...	3.0	40.0
4	False	False	...	28.0	25.0
5	False	False	...	4.0	39.0
6	False	True	...	17.0	68.0
7	False	True	...	3.0	38.0
8	True	False	...	11.0	59.0
9	False	False	...	6.0	49.0

	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	\
0	04B	1144606.0	1903566.0	2015	02/10/2018 03:50:01 PM	
1	26	1166468.0	1860715.0	2015	02/10/2018 03:50:01 PM	
2	08B	1185075.0	1875622.0	2015	02/10/2018 03:50:01 PM	
3	08B	1178033.0	1870804.0	2015	02/10/2018 03:50:01 PM	
4	03	1144920.0	1898709.0	2015	02/10/2018 03:50:01 PM	
5	08B	1183018.0	1872537.0	2015	02/10/2018 03:50:01 PM	
6	08B	1170859.0	1858210.0	2015	02/10/2018 03:50:01 PM	
7	08B	1178746.0	1876914.0	2015	02/10/2018 03:50:01 PM	
8	18	1164279.0	1880656.0	2015	02/10/2018 03:50:01 PM	
9	08B	1179637.0	1840444.0	2015	02/10/2018 03:50:01 PM	

	Latitude	Longitude	Location
0	41.891399	-87.744385	(41.891398861, -87.744384567)
1	41.773372	-87.665319	(41.773371528, -87.665319468)
2	41.813861	-87.596643	(41.81386068, -87.596642837)
3	41.800802	-87.622619	(41.800802415, -87.622619343)
4	41.878065	-87.743354	(41.878064761, -87.743354013)
5	41.805443	-87.604284	(41.805443345, -87.604283976)
6	41.766403	-87.649296	(41.766402779, -87.649296123)
7	41.817553	-87.619819	(41.817552577, -87.619818523)
8	41.828138	-87.672782	(41.828138428, -87.672782106)
9	41.717455	-87.617663	(41.71745472, -87.617663257)

[10 rows x 22 columns]

```
In [4]: map_hooray = folium.Map(location=[41.8, -87.6],
                                   zoom_start = 10)
```

```
df_heat = df[df['Year'] == 2015]
df_heat = df_heat[df_heat['Primary Type'] == 'BATTERY']
df_heat = df_heat[df_heat['Arrest'] == True]
df_heat = df_heat.dropna()
# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray
```

```
Out[4]: <folium.folium.Map at 0x7f3bfe2edfd0>
```

```
In [5]: print(df['Primary Type'].value_counts())
```

THEFT	1367778
BATTERY	1193225
CRIMINAL DAMAGE	749597
NARCOTICS	700715
OTHER OFFENSE	405280
ASSAULT	402476
BURGLARY	378983
MOTOR VEHICLE THEFT	306967
ROBBERY	248401
DECEPTIVE PRACTICE	246870
CRIMINAL TRESPASS	188115
PROSTITUTION	67738
WEAPONS VIOLATION	66423
PUBLIC PEACE VIOLATION	46718
OFFENSE INVOLVING CHILDREN	43402
CRIM SEXUAL ASSAULT	25684
SEX OFFENSE	24082
GAMBLING	14231
INTERFERENCE WITH PUBLIC OFFICER	14105
LIQUOR LAW VIOLATION	13845
ARSON	10867
HOMICIDE	8966
KIDNAPPING	6538
INTIMIDATION	3796
STALKING	3213
OBSCENITY	510
CONCEALED CARRY LICENSE VIOLATION	160
PUBLIC INDECENCY	150
NON-CRIMINAL	135
OTHER NARCOTIC VIOLATION	123
NON - CRIMINAL	38
HUMAN TRAFFICKING	37
RITUALISM	23
NON-CRIMINAL (SUBJECT SPECIFIED)	6
DOMESTIC VIOLENCE	1

Name: Primary Type, dtype: int64

```
In [6]: map_hooray = folium.Map(location=[41.8, -87.6],  
                                   zoom_start = 10)
```

```
df_heat = df[df['Primary Type'] == 'HOMICIDE']  
df_heat = df_heat[df_heat['Arrest'] == True]  
df_heat = df_heat[df_heat['Year'] == 2001]
```

```

df_heat = df_heat.dropna()
# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray

```

Out[6]: <folium.folium.Map at 0x7f3c28956278>

```

In [7]: map_hooray = folium.Map(location=[41.8, -87.6],
                                zoom_start = 10)

```

```

df_heat = df[df['Primary Type'] == 'PROSTITUTION']
df_heat = df_heat[df_heat['Arrest'] == True]
df_heat = df_heat[df_heat['Year'] == 2001]
df_heat = df_heat.dropna()
# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray

```

Out[7]: <folium.folium.Map at 0x7f3c289568d0>

```

In [8]: map_hooray = folium.Map(location=[41.8, -87.6],
                                zoom_start = 10)

```

```

df_heat = df[df['Primary Type'] == 'PROSTITUTION']
df_heat = df_heat[df_heat['Arrest'] == True]
df_heat = df_heat[df_heat['Year'] == 2015]
df_heat = df_heat.dropna()
# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray

```

Out[8]: <folium.folium.Map at 0x7f3c286fea20>

```

In [24]: map_hooray = folium.Map(location=[41.8, -87.6],
                                zoom_start = 10)

```

```

df_heat = df[df['Primary Type'] == 'PROSTITUTION']
df_heat = df_heat[df_heat['Arrest'] == True]
df_heat = df_heat[df_heat['Year'] == 2017]
df_heat = df_heat.dropna()

```

```

# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray

```

Out[24]: <folium.folium.Map at 0x7f3bdf517a20>

In [10]: print(df.columns.values)

```

['ID' 'Case Number' 'Date' 'Block' 'IUCR' 'Primary Type' 'Description'
 'Location Description' 'Arrest' 'Domestic' 'Beat' 'District' 'Ward'
 'Community Area' 'FBI Code' 'X Coordinate' 'Y Coordinate' 'Year'
 'Updated On' 'Latitude' 'Longitude' 'Location']

```

In [25]: map_hooray = folium.Map(location=[41.8, -87.6],
zoom_start = 10)

```

#df_heat = df[df['Primary Type'] == 'ASSAULT']
#df_heat = df[df['Block'] == '001XX N STATE ST']
df_heat = df[df['Year'] == 2001]
df_heat = df_heat.dropna()
# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray

```

Out[25]: <folium.folium.Map at 0x7f3bdbbace198>

In [30]: map_hooray = folium.Map(location=[41.8, -87.6],
zoom_start = 10)

```

#df_heat = df[df['Primary Type'] == 'ASSAULT']
df_heat = df[df['Block'] == '100XX W OHARE ST']
df_heat = df_heat[df_heat['Year'] == 2017]
df_heat = df_heat.dropna()
# List comprehension to make out list of lists
heat_data = [[row['Latitude'],row['Longitude']] for index, row in df_heat.iterrows()]

# Plot it on the map
HeatMap(heat_data).add_to(map_hooray)
map_hooray

```

Out[30]: <folium.folium.Map at 0x7f3bd6753f98>

```

In [ ]: small_df = df[df['Year'] == 2001]
        g = small_df['Primary Type'].hist(bins=50, figsize=(40,10))
        plt.show()

In [34]: print(df['Date'][0].split('/'))

['03', '18', '2015 07:44:00 PM']

In [ ]: print(df[df['Year'] == 2005].Block.value_counts())

In [44]: def get_month(datestring):
        return datestring.split('/')[0]

In [45]: df['Month'] = df['Date'].apply(get_month)
        print(df.columns.values)

['ID' 'Case Number' 'Date' 'Block' 'IUCR' 'Primary Type' 'Description'
 'Location Description' 'Arrest' 'Domestic' 'Beat' 'District' 'Ward'
 'Community Area' 'FBI Code' 'X Coordinate' 'Y Coordinate' 'Year'
 'Updated On' 'Latitude' 'Longitude' 'Location' 'Month']

In [46]: def get_day(datestring):
        return datestring.split('/')[1]

In [47]: df['Day'] = df['Date'].apply(get_day)
        print(df.columns.values)

['ID' 'Case Number' 'Date' 'Block' 'IUCR' 'Primary Type' 'Description'
 'Location Description' 'Arrest' 'Domestic' 'Beat' 'District' 'Ward'
 'Community Area' 'FBI Code' 'X Coordinate' 'Y Coordinate' 'Year'
 'Updated On' 'Latitude' 'Longitude' 'Location' 'Month' 'Day']

In [ ]: print(df['Month'].value_counts())

In [ ]: fig, ax = plt.subplots()
        state_st_df = df[df['Block'] == "001XX N STATE ST"]
        g = state_st_df['Primary Type'].hist(bins=50, figsize=(40,10))
        plt.show()

In [ ]: fig, ax = plt.subplots()
        state_st_df = df[df['Block'] == "100XX W OHARE ST"]
        g = state_st_df['Primary Type'].hist(bins=50, figsize=(40,10))
        plt.show()

In [ ]: small_df = df[df['Year'] == 2001]
        small_df['Primary Type'].value_counts
        '''
        g = small_df['Primary Type'].hist(bins=50, figsize=(40,10))
        plt.show()
        '''

```